Subject: what is an efficient lossless compression way to store a gray-scale image
Posted by xje4e on Tue, 26 Aug 2003 14:52:21 GMT
View Forum Message <> Reply to Message

Hi, there,

For a gray-scale image, I haven't found a good way to store it yet. It
seems that there is no way to store gray-scale images in IDL
efficiently. If so, that will be too bad, because all the satellite
images we processed are very large and occupy a lot of space.

I tried 'tiff' format with Packbits compression, but it does not help
for gray-scale image. Sometimes the file size of the compressed image
is even larger than the original raw data!!!

Can anyone give me some hint? Thank you very much!

Regards,

Julia

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by R.G. Stockwell on Tue, 26 Aug 2003 16:42:07 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message
news:10ea38a6.0308260652.6a1e1b9a@posting.google.com...
> Hi, there,
>
> For a gray-scale image, I haven't found a good way to store it yet. It
> seems that there is no way to store gray-scale images in IDL
> efficiently. If so, that will be too bad, because all the satellite
> images we processed are very large and occupy a lot of space.
>
> I tried 'tiff' format with Packbits compression, but it does not help
> for gray-scale image. Sometimes the file size of the compressed image
> is even larger than the original raw data!!!
>
> Can anyone give me some hint? Thank you very much!
>
> Regards,
>
> Julia

Hi Julia,

this is not trying to be a flippant reply, but the easiest solution is to
simply buy more hardrives.
A couple of 200G drives fore a couple hundred each might solve the problem.

Or, offhand I would say the best you can do is probably directly writing a
binary file
of the appropriate precision (and use the compress keyword on the openw
procedure).
For instance, if you have 16 bit numbers, write an array of integers.
A quick look seems to show that your tiff NONcompressed files are pretty
good.
(of course, the compression you actually gets depends on the data)

I was curious, so I made a little example.
here, the data is 1024 x 1024 16 bit integers, so it should
be about 2Megs in size 2,097,152 bytes.

```
len = 1024
randomdata = fix(100*randomn(seed,len,len))
regulardata = indgen(len,len)
openw,lun,'randomdata_compress.dat',/get_lun,/compress
 writeu,lun,randomdata
free_lun,lun
openw,lun,'randomdata.dat',/get_lun
 writeu,lun,randomdata
free_lun,lun
openw,lun,'regulardata_compress.dat',/get_lun,/compress
 writeu,lun,regulardata
free_lun,lun
openw,lun,'regulardata.dat',/get_lun
 writeu,lun,regulardata
free_lun,lun
write_tiff,'tiff_compress_random',randomdata,compression=2
 write_tiff,'tiff_compress_regular',regulardata,compression=2
write_tiff,'tiff_regular',regulardata,compression=0
```

These commands give the following file sizes:

```
08/26/2003  10:36a         1,477,230 randomdata_compress.dat
08/26/2003  10:36a         2,097,152 randomdata.dat
08/26/2003  10:36a         1,905,228 regulardata_compress.dat
08/26/2003  10:36a         2,097,152 regulardata.dat
08/26/2003  10:31a         1,058,072 tiff_compress
08/26/2003  10:36a         1,058,072 tiff_compress_random
08/26/2003  10:36a         1,058,062 tiff_compress_regular
08/26/2003  10:36a         1,049,862 tiff_regular
```

So, the tiff command is actually pretto good, giving you a ~50% size, and it works
beter than the gzip compression in the openw command.
So perhaps your best bet is just to buy more disk space, or reduce your data based on
some other criteria (i.e. bin the data or downsample to a larger sampling size in space, or bin/downsample
in time if that is an appropiate for your applications)

Cheers,
bob

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by Jonathan Boswell on Tue, 26 Aug 2003 17:23:01 GMT
View Forum Message <> Reply to Message

Julia wrote:
> seems that there is no way to store gray-scale images in IDL
> efficiently...

I believe GIF is lossless if you have fewer than 256 "colors". What version of IDL do you have? There are licensing issues pertaining to the LZW compression used by the GIF standard, and recent versions of IDL will not run WRITE_GIF w/o additional expense.

- JB

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by xje4e on Tue, 26 Aug 2003 17:23:45 GMT
View Forum Message <> Reply to Message

First I forgot to mention, the image is not necessary byte or integer.
It could be float type. So .png file is not suitable in some cases.
Anyway, it works great in 8 or 16 bits cases.

> Or, offhand I would say the best you can do is probably directly writing a
> binary file of the appropriate precision (and use the compress keyword on the >openw procedure).
Thanks for your suggestion! But if we wrote the files in .dat rather than the image files. Then we need to rewrite it into an image file if we want to look at the image. So not a good idea.

> I was curious, so I made a little example.
> here, the data is 1024 x 1024 16 bit integers, so it should
> be about 2Megs in size 2,097,152 bytes.
> write_tiff,'tiff_compress_random',randomdata,compression=2
> write_tiff,'tiff_compress_regular',regulardata,compression=2
> write_tiff,'tiff_regular',regulardata,compression=0
>
>
> These commands give the following file sizes:
>
> 08/26/2003 10:36a   1,477,230 randomdata_compress.dat
> 08/26/2003 10:36a   2,097,152 randomdata.dat
> 08/26/2003 10:36a   1,905,228 regulardata_compress.dat
> 08/26/2003 10:36a   2,097,152 regulardata.dat
> 08/26/2003 10:31a   1,058,072 tiff_compress
> 08/26/2003 10:36a   1,058,072 tiff_compress_random
> 08/26/2003 10:36a   1,058,062 tiff_compress_regular
> 08/26/2003 10:36a   1,049,862 tiff_regular
>
> So, the tiff command is actually pretto good, giving you a ~50% size.
I think tiff command without compression will keep the original size
2.097.152bytes. Because the data is integer type, so when writing the
tiff, /uint is used. What you got is byte type image data.

Regards,

Julia

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by R.G. Stockwell on Tue, 26 Aug 2003 17:55:34 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message
news:10ea38a6.0308260923.28c8cad9@posting.google.com...
> First I forgot to mention, the image is not necessary byte or integer.
> It could be float type. So .png file is not suitable in some cases.
> Anyway, it works great in 8 or 16 bits cases.

> Thanks for your suggestion! But if we wrote the files in .dat rather
> than the image files. Then we need to rewrite it into an image file if
> we want to look at the image. So not a good idea.

You could always whip up an IDL GUI to read and display the images.

...

>> So, the tiff command is actually pretto good, giving you a ~50% size.
> I think tiff command without compression will keep the original size
> 2.097.152bytes. Because the data is integer type, so when writing the
> tiff, /uint is used. What you got is byte type image data.

Doh! You are right, my error.

One thing that comes to mind is to hold the data in the type of variables
that are coming from the instrument that created the images. For instance,
if these are CCD images they will digitized at a certian level of precision
(i.e. 8, 10, 12, or 16 bit Analog to Digital conversion). You can make
sure you
keep the data files at this precision (notably, 12 bit images will be stored
inefficiently
in IDL normal types as a float or as an integer).
There is potentlially some room there to improve efficiency. This may
require that you
"undo" the scaling of the data, and for a lot of data, this may not really
be possible
(i.e. Dopplergrams of the sun's surface for instance).

So, if you can't make the above changes, my guess is that a 50% compression
is about as good as you are going to get in general, and I don't know of any
way to improve
on the compression routines that you can use with the write_tiff. (while
being lossless).

Other image options, You can do *.fits images, which can be read in IDL by
some user routines,
and also with other software like ds9 on linux boxes.

Cheers,
bob

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by Liam E. Gumley on Tue, 26 Aug 2003 21:24:24 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message
news:10ea38a6.0308260652.6a1e1b9a@posting.google.com...
> Hi, there,
>
> For a gray-scale image, I haven't found a good way to store it yet. It
> seems that there is no way to store gray-scale images in IDL
> efficiently. If so, that will be too bad, because all the satellite
> images we processed are very large and occupy a lot of space.

>
> I tried 'tiff' format with Packbits compression, but it does not help
> for gray-scale image. Sometimes the file size of the compressed image
> is even larger than the original raw data!!!
>
> Can anyone give me some hint? Thank you very much!

If you use the OPEN routine to open the files for reading and writing, then
use the COMPRESS keyword to activate GZIP compression:

"If COMPRESS is set, IDL reads and writes all data to the file in the
standard GZIP format. IDL's GZIP support is based on the freely available
ZLIB library version 1.1.3 by Mark Adler and Jean-loup Gailly. This means
that IDL's compressed files are 100% compatible with the widely available
gzip and gunzip programs. COMPRESS cannot be used with the APPEND keyword."

This method will automatically compress/decompress any data type you wish to
read or write in IDL.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by George N. White III on Wed, 27 Aug 2003 22:52:02 GMT
View Forum Message <> Reply to Message

On Tue, 26 Aug 2003, Xiaoying Jin wrote:

> Hi, there,
>
> For a gray-scale image, I haven't found a good way to store it yet. It
> seems that there is no way to store gray-scale images in IDL
> efficiently. If so, that will be too bad, because all the satellite
> images we processed are very large and occupy a lot of space.
>
> I tried 'tiff' format with Packbits compression, but it does not help
> for gray-scale image. Sometimes the file size of the compressed image
> is even larger than the original raw data!!!
>
> Can anyone give me some hint? Thank you very much!

In general, rather than creating image for viewing data, it is preferrable
to preserve the original data and provide tools to view the data
"directly". You may, however, want some low-resolution "browse" images.

By viewing the actual data files you reduce the disk space devoted to multiple copies of the same information in different formats, but also the problems that arise in trying to interprete images when you don't have ready access to the actual data values or geographical coordinates.

In retrospect, the smart approach would have been to define satellite data formats (e.g. TDF, HDF) as a "subset" of TIFF and then provide Photoshop plugins. This would have satisfied the people who "just want some images" to look at".

Some compression methods require a non-free license, but the PNG format uses a "free" algorithm. You can use an image format without compression or write a raw data file and then apply one of the many file compression tools (zip, gzip, etc.). IDL has a /compress keyword that supports gzip compression, but (on a decent OS) you can use other methods by writing data to a named pipe "connected" to an external compression program.

How much compression you get depends on the data, so it may pay to run tests on your images. Satellite images are often noisy. You may get 12 bits-per-pixel (bpp), but if the bottom 4 bits are random the file won't compress well. In that case, converting the file to 8 bpp may not loose much information, but would substantially decrease the raw file size and producing better compression.

--
George N. White III <gnw3@acm.org>
